Detailed project description: Astrophysical turbulence and dynamo action

(Nordita, Stockholm)

October 11, 2024

1 Overview

In the astrophysics group at Nordita, we are routinely using the Pencil Code[1], which is hosted by Github[2]. Over the past few years, this code has been extended to be accelerated by GPUs. The Pencil Code group in Finland has been developing it on LUMI, and more recently, we also have benchmarked the code on Dardel. The speedup is up to a factor 16 for problems of practical interest. For a subset of our research, the GPU acceleration has not yet been implemented. Therefore, we also request standard CPU-based time. Also, at least on Dardel, the total problem size that we can simulate with GPU acceleration may not yet be large enough, which is another reason why we also need time on CPU nodes.

The Pencil Code is an open-source code developed by Brandenburg, his current and former coworkers, some of whom are part of this project, as well as others that have been invited to join the effort. The performance of this code has been discussed at several international conferences. The code has been optimized over the years and is still being improved in terms of performance and new features are also being added. All of the 37,375 revisions since 2001 are publicly available through our repository. We have adapted and optimized this code for spherical polar coordinate system. This addition to the code is used in several of the problems discussed here. The code runs well on all the different platforms.

2 Resource usage

We begin by reviewing the performance of the GPU-accelerated simulations with the Pencil Code using both LUMI and Dardel.

2.1 GPU-acceleration on the Pencil Code

All the scaling results reported here are based on three-dimensional simulations of decaying hydromagnetic turbulence, just as in the recent paper by Brandenburg et al. (2024). Here, we compare the Pencil Code (PC) on CPUs against the PC with Astaroth[3] embedded (PC-A), where Astaroth integrates the partial differential equations of magnetohydrodynamics while the PC performs all peripheral tasks (diagnostics & I/O). On Dardel, we build with `gfortran/gcc` and `nvcc` using the modules:

```
gcc/12.2.0   craype-accel-amd-gfx90a  PDC/23.12
PrgEnv-gnu/8.5.0  cray-mpich/8.1.27   rocm/5.7.0
```

For the timing results, we routinely output with the PC both the wallclock and normalized times per time step and mesh point. *On Dardel*, we performed the following set of tests: PC-A, 8GPUs = 1 node, grid size 512^3, three repetitions:

```
Wall clock time/timestep/meshpoint [microsec] = 1.9076658E-03
Wall clock time/timestep/meshpoint [microsec] = 2.3275883E-03
Wall clock time/timestep/meshpoint [microsec] = 1.9224858E-03
```

[1] `http://www.nordita.org/software/pencil-code` (The Pencil Code Collaboration, 2021)
[2] `https://github.com/pencil-code`
[3] `https://bitbucket.org/jpekkila/astaroth/`

Thus, the normalized times per time step and mesh point are between 1.9 and 2.3 nanoseconds. For completeness, we also indicate here the actual wallclock times for the same three runs:

```
Wall clock time [hours] =  7.119E-02 (+/-  5.5556E-12)
Wall clock time [hours] =  8.687E-02 (+/-  5.5556E-12)
Wall clock time [hours] =  7.175E-02 (+/-  8.3333E-12)
```

PC, 64 CPUs = 1 node, grid size 512^3, two repetitions:

```
Wall clock time/timestep/meshpoint [microsec] = 2.2191811E-02
Wall clock time/timestep/meshpoint [microsec] = 2.2515382E-02
```

Thus, without GPU acceleration, the normalized times per time step and mesh point are between 22 and 23 nanoseconds, so the comparison reveals a *speedup of about 11*. For testing weak scaling, we also performed a problem that is 8 times larger using 8 nodes instead of 1:
PC-A, 64GPUs = 8 nodes, grid size 1024^3

```
Wall clock time [hours] =  7.786E-02 (+/-  5.5556E-12)
Wall clock time/timestep/meshpoint [microsec] = 2.6079823E-04
```

Thus, the normalized time per time step and mesh point is 0.26 nanoseconds. Here we only did one such test. Therefore, the comparison reveals roughly the same wall-clock time as with PC-A on one node and the same grid size per process. This is therefore very satisfactory.
On LUMI, we performed an analogous comparison:
PC-A, 1 node, 8 GPUs, grid size 380^3,

```
Wall clock time [hours] =  1.522E-02 (+/-  8.3333E-12)
Wall clock time/timestep/meshpoint [microsec] = 1.1079179E-03
```

Thus, the normalized time per time step and mesh point is 1.1 nanoseconds.
PC, 1 node, 64 CPUs, grid size 352^3

```
Wall clock time [hours] =  0.192     (+/-  5.5556E-12)
Wall clock time/timestep/meshpoint [microsec] = 1.7558866E-02
```

Thus, the normalized time per time step and mesh point is 18 nanoseconds, so the *speedup is about 16*.
Next, we report weak scaling results:
PC-A, 1 node, 8 GPUs, grid size 512^3,

```
Wall clock time [hours] =  4.535E-02 (+/-  5.5556E-12)
Wall clock time/timestep/meshpoint [microsec] = 1.2150889E-03
```

PC-A, 8 nodes, 64 GPUs, grid size 1024^3,

```
Wall clock time [hours] =  1.261E-02 (+/-  5.5556E-12)
Wall clock time/timestep/meshpoint [microsec] = 4.2233689E-05
```

PC-A, 2048^3, 64 nodes, 512 GPUs

```
Wall clock time [hours] =  1.510E-02 (+/-  5.5556E-12)
Wall clock time/timestep/meshpoint [microsec] = 6.3206880E-06
```

The last two runs show roughly equal wallclock time. The longer time for the smallest grid size remains unclear.

2.2 CPU simulations with the Pencil Code

On Dardel, we run production runs with up to 2048^3 mesh points on up to 32768 cores. A typical run requires at least 500,000 time steps, but it can sometimes be much more, depending on circumstances. With $4.2\times10^{-4}\mu$s per meshpoint and per timestep on Dardel. We anticipate that the time will be about 4 days of wall clock time at a cost of 600,000 CPU hours, while with $3.5\times10^{-3}\mu$s per meshpoint and per timestep, this means less than 3 days of wall clock time at a cost of 30,000 CPU hours per run.

To address properly the critical question of the dependence on the magnetic Reynolds number we have to use high resolution runs. As we move from 256^3 and 512^3 to 2048^3 mesh points (and correspondingly higher magnetic Reynolds numbers), we see the emergence of small-scale dynamo action at all depth. This does not yet affect the 512^3 runs, but for the 2048^3 run, small-scale dynamo action becomes critical. The last of these runs is for a deeper domain, so as to include more safely the deep parts where it is important to reach values of the mean field normalized to the equipartition value below 0.01, but this appears not to be possible due to small-scale dynamo action.

To confirm our ideas and to understand the effects of small-scale dynamo action, we plan to perform about 2 big runs per month on Dardel, which requires at least 1,000,000 CPU hours, and about 4 intermediate ones, which requires 150,000 CPU hours on each of them. To shed light on some of the observational features of accretion disks, particularly those related to interactions between the inflow stream and the accretion disk, we must run high-resolution global simulations with radiation. This is a new activity, where each run takes about 300,000 CPU hours. We plan to run three of those. Our total time requirement is therefore 2,500,000 CPU hours on Dardel and Tetralith combined.

Computationally, all machines are comparable, but there can be unexpected future changes or outages on some machines that hamper scientific progress. Important is also the waiting time in the queue and occasional opportunities when jobs start immediately. Most of our activity will reside on Dardel. However, to maximize our scientific productivity, and not to be affected too much by outages and long waiting times, we also apply for time on Tetralith.

Regarding scaling tests, we have previously determined strong scaling of Pencil code on Triolith for three mesh sizes. The time per time step and mesh point is given for different processor numbers and layouts. Generally, it is advantageous to keep the number of processors in the x direction small. The code is well adapted to modern computing platforms.

Performancewise, Cray with O2 optimization is equivalent to gnu with O3. While gnu-O3 is able to handle memory or whatever compiler problems much better, it is otherwise not better than Cray-O2, and often some 10–20% slows, but this is within the measurement accuracy; see `https://github.com/pencil-code/website/blob/master/NewsLetters/2021/newsletter3.pdf`. More details can be found on `https://github.com/pencil-code/pencil-code/tree/master/doc/timings`.

3 Scientific challenges

Inverse cascading in non-helical MHD. Today, high resolution numerical turbulence simulations are feasible. They allow for fairly large turbulent inertial ranges, but not all the resources must be spent on a large inertial range. It is also important to allow for enough scale separation toward very long length scales to avoid artifacts from a finite domain size. To help optimizing the computational resources, various groups have employed numerical tools to approach the physically relevant regime: (i) reduce the viscous and diffusive subranges by adopting hyperviscosity and magnetic hyperdiffusivity, and (ii) to make the viscous and magnetic diffusion coefficients time-dependent, or do use, for example, slope-limited diffusion to prevent the code from crashing at early times when the turbulence is extremely vigorous and to allow for the physical viscosity to be as small as possible at late times. All these tools imply small artifacts that we need to examine very carefully. The slope-limited diffusion technique has recently been used in some of our simulations of chiral MHD. To assess the role of artifacts in simulations, we now need to perform more detailed studies by monitoring the evolutionary tracks in a pq diagram for different combinations of such numerical tools. We will therefore perform divers surveys to extract physically meaningful results from artifacts. This part will mostly be carried out with GPU-acceleration. However, some of the calculations with radiation transport will use CPUs for now.

Inflationary magnetogenesis and gravitational wave production. In two recent papers with Ramkishor Sharma, a new Nordita postdoctoral researcher, we have performed new simulations of inflationary magnetogenesis and gravitational wave production. In those models, we solved the equations for electromagnetic waves during the post-inflationary phase (the reheating phase), and switched to magnetohydrodynamics at the beginning of the radiation-dominated phase. In future models, we will include equations describing the evolution of scalar potentials in cosmology. This will lead to more realistic models. All models are three-dimensional and typical resolutions are of the order of 1024^3 to 2048^3 mesh points. This part will mostly be carried out with GPU-acceleration, although we have not yet tested the gravitational wave module with GPUs yet.

Gravitational wave polarization. We study the influence of helical magnetic fields on the production of gravitational waves. Gravitational waves provide an as yet unexplored window into the earliest moments of the Big Bang, not obscured by the last scattering surface given by the hitherto studied cosmic microwave background. The production of gravitational radiation from cosmological turbulence was calculated analytically by Kosowsky et al. (2002) and Gogoberidze et al. (2007). Helical magnetic fields produce non-vanishing cross-polarization in the gravitational wave spectrum (Kahniashvili et al., 2005; Caprini & Durrer, 2006), which would be observable with LISA. Hindmarsh et al. (2017) have recently presented detailed numerical models of gravitational waves from phase transition nucleation bubbles produced during the electroweak phase transition. Our new work involves the calculation of gravitational waves using the Pencil Code, where a gravitational wave solver has already been successfully implemented. This part will mostly be carried out with GPU-acceleration, although we have not yet tested the gravitational wave module with GPUs yet.

Chiral MHD. The chiral magnetic effect leads to a current along a magnetic field if the number of left- and right-handed Fermions is unequal. This effect has received significant attention in just the last few years. We are now able for the first time to perform a comprehensive study of the chiral magnetic effect in real turbulence. Earlier theoretical studies applied to neutron stars and the early Universe did not result in realistic estimates for the turbulence. Thus, the use of simulations is absolutely critical to making significant progress. Our recent work on the early Universe has brought us a significant step forward. We will now focus on neutron stars, which may have several important advantages. First, only one sign of chirality will be produced. Second, the timescales are short, giving us ample time for the subsequent inverse cascade to yield large length scales. Together with the helicity produced from rotation and stratification, the end result may produce a realistic model of observed pulsars. This part will mostly be carried out with GPU-acceleration, although we have not yet tested the chiral MHD module with GPUs yet. We expect that this will not cause any problems.

Entropy rain convection. Convection is a highly nonlocal phenomenon where low entropy blobs can descent over large depths through the convection zone. This causes an extra contribution to the enthalpy flux that is not described by a local entropy gradient, but by the value of the mean squared entropy fluctuation relative to the background. This always transports energy outward. The resulting stratification may end up being slightly subadiabatic instead of being slightly superadiabatic and therefore host gravity modes that may be observationally detectable. Corresponding simulations were produced by Brandenburg (2016) and Käpylä et al. (2017), but the effect of rotation is not well understood. This will be studied using upcoming simulations. The use of radiation transport precludes at the moment GPU acceleration and will therefore use CPUs.

Solar/stellar dynamo simulations. Local simulations will be used to develop what we call smart boundary conditions for application in the global simulations. The purpose of such boundary conditions is to compactify the small-scale physics of the surface-driving layer in order to control the global simulations, which cannot resolve these scales, in a physically realistic way. Here we assume that (i) stellar turbulence is essentially driven by cooling in the surface-driving layer and (ii) large-scale structures like giant cells or a global dynamo field would not markedly affect the overall properties of the convection. Then, local Cartesian boxes, which extend vertically just deep enough so that the (non-physical) boundary conditions to be applied at their bottom have no significant effect on the near-surface convection (say, 30 Mm deep)

and which are horizontally just wide enough to capture the essential topology and dynamics of the granulation, will be employed to solve the full convection problem with the necessary high grid resolution (say, 100 Mm horizontal extent) and with physically meaningful boundary conditions at their top. Time series of the simulated physical quantities on a horizontal plane placed at the estimated bottom of the surface-driving layer inside the computational box will be employed to define the boundary conditions at the top of a global simulation model which extends from the bottom of the convection zone (say 200 Mm depth) with physically meaningful boundary conditions to the bottom of the surface-driving layer. A simple way of doing this consists in directly employing the quantities from the local-box simulations as Dirichlet boundary conditions of the global model. Due to its coarser resolution, the data have to be properly restricted. As the simulated model time interval of the global model will in general be much longer than the one of the local model, the problem arises of how the boundary values should be repeatedly used without introducing a strict periodicity. This approach will allow incorporating the NSSL in the global simulations, without needing to resolve the surface layer in one and the same model. A number of modules preclude GPU acceleration at the moment and we will therefore use CPUs.

4 Research group and management

The work in the astrophysics group at Nordita covers a broad range of topics from kinetic simulations over gravitational wave physics and the early universe to solar physics and meteorology. Our research group consists currently of the following people:

Kyrylo Bondarenko (Nordita fellow)
Dr Oksana Iarygina (Marie Curie fellow)
Mr Gustav Larsson (Master student)
Dr Lars Mattsson (guest researcher)
Dr Dhrubaditya Mitra (assistant professor)
Mr Patrik Tengnér (Master student)

The monthly usage within the group is monitored and discussed during our weekly group meetings.

References

Brandenburg A., "Stellar mixing length theory with entropy rain," *Astrophys. J.* **832**, 6 (2016).

Brandenburg, A., Neronov, A., & Vazza, F., "Resistively controlled primordial magnetic turbulence decay," *Astron. Astrophys.* **687**, A186 (2024).

Caprini, C., & Durrer, R., "Gravitational waves from stochastic relativistic sources: Primordial turbulence and magnetic fields," *Phys. Rev. D* **74**, 063521 (2006).

Gogoberidze, G., Kahniashvili, T., & Kosowsky, A., "Spectrum of gravitational radiation from primordial turbulence," *Phys. Rev. D* **76**, 083002 (2007).

Hindmarsh, M., Huber, S. J., Rummukainen, K., & Weir, D. J.: 2017, "Shape of the acoustic gravitational wave power spectrum from a first order phase transition," arXiv:1704.05871

Kahniashvili, T., Gogoberidze, G., & Ratra, B., "Polarized Cosmological Gravitational Waves from Primordial Helical Turbulence," *Phys. Rev. Lett.* **95**, 151301 (2005).

Käpylä, P. J., Rheinhardt, M., Brandenburg, A., Arlt, R., Käpylä, M. J., Lagg, A., Olspert, N., & Warnecke, J., "Extended subadiabatic layer in simulations of overshooting convection," *Astrophys. J. Lett.* **845**, L23 (2017).

Kosowsky, A., Mack, A., & Kahniashvili, T., "Gravitational radiation from cosmological turbulence," *Phys. Rev. D* **66**, 024030 (2002).

Pencil Code Collaboration: Brandenburg, A., Johansen, A., Bourdin, P. A., Dobler, W., Lyra, W., Rheinhardt, M., Bingert, S., Haugen, N. E. L., Mee, A., Gent, F., Babkovskaia, N., Yang, C.-C., Heinemann, T., Dintrans, B., Mitra, D., Candelaresi, S., Warnecke, J., Käpylä, P. J., Schreiber, A., Chatterjee, P., Käpylä, M. J., Li, X.-Y., Krüger, J., Aarnes, J. R., Sarson, G. R., Oishi, J. S., Schober, J., Plasson, R., Sandin, C., Karchniwy, E., Rodrigues, L. F. S., Hubbard, A., Guerrero, G., Snodin, A., Losada, I. R., Pekkilä, J., & Qian, C., "The Pencil Code, a modular MPI code for partial differential equations and particles: multipurpose and multiuser-maintained," *J. Open Source Softw.* **6**, 2807 (2021).